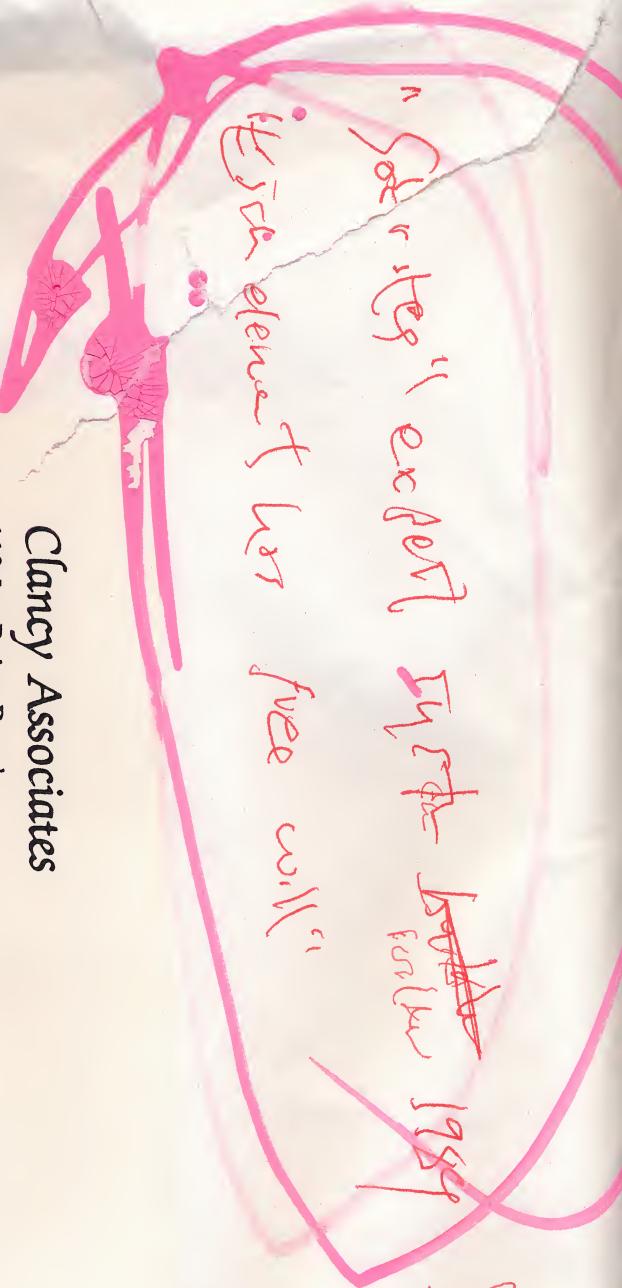


~~DISC Enclosure~~

"Sorcery" expert in the ~~bottle~~  
it's element free will



Clancy Associates  
110 La Bolsa Road  
Walnut Creek, CA 94598

MARLENE MALLICOAT  
2250 GREEN STREET  
SAN FRANCISCO, CA 94123

R.  
R.  
R.  
R.  
R.  
R.  
R.

**3M**

SOCRATES

DS, HD PC/AT COMPATIBLE  
double side, high density



SOCRATES by

#2

Software Technology

**3M** diskettes

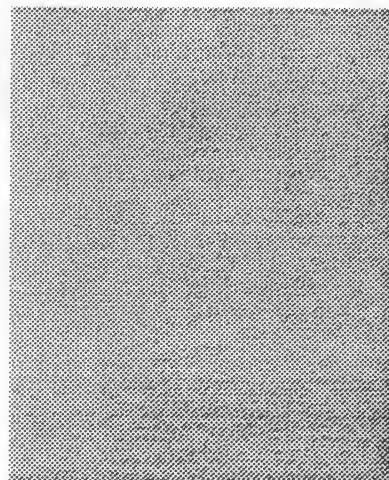
TED NELSON  
[Swiss]  
1989

SOKRATES MANUAL  
*April 1989*

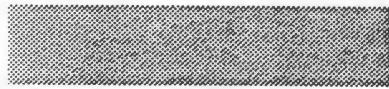
# SOFTWARE TECHNOLOGY

USER GUIDE MANUAL - 1.00

PAGE 1



**SOKRATES<sup>TM</sup>**



EXPERT  
APPLICATIONS  
GENERATOR  
*5th Generation*  
with  
Relational Informations  
System



Reference manual

Version 1.00

Copyright ©1988, 1989  
All right reserved

SOFTWARE TECHNOLOGY  
Manoah , CH -1528 Surpierre

# SOFTWARE TECHNOLOGY

## TABLE OF CONTENTS

<b>1. INTRODUCTION</b>	<b>Page 3</b>
<b>2. INSTALLATION OF SOKRATES</b>	<b>Page 6</b>
<b>3. SOKRATES EXPERT SYSTEM</b>	<b>Page 10</b>
Presentation	Page 10
General Dictionary	Page 10
Object approach concept	Page 12
OBJECTS	Page 14
Infos	Page 14
File	Page 16
Extention	Page 18
Functions	Page 19
Sheet	Page 20
Tasks	Page 23
What is a task ?	Page 25
Types of tasks	Page 29
Controlled breaks	Page 30
Links	Page 31
<b>4. SYNTAX</b>	<b>Page 35</b>
Operators	Page 35
Operands	Page 36
Commands	Page 38
<b>5. SCENARIOS FOR IMPLEMENTATION</b>	<b>Page 75</b>
Introduction	Page 75
Scenarios of development	Page 77
<b>6. GLOSSARY</b>	<b>Page 78</b>

*This manual is written as support for the Sokrates courses and addresses application developers who follow the course: Database Conception and Design with Sokrates.*

*No part of it may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without Software Technology (Switzerland) prior written consent.*

*SOKRATES is a ST product*

# SOFTWARE TECHNOLOGY

## 1 - INTRODUCTION

SOKRATE is a professional Expert System Generator of Applications of the 5th generation. It features a powerful relational database.

It allows the implementation of professional and sophisticated applications without using a programming language.

It does not generate sources, nor compile objects or produce programmes.

Although it is written in "C Language", Its users do not have to know any programming language. They develop and dialogue, with SOKRATES, in their own language.

SOKRATES stands apart of all similar products for the following aspects :

1.The Generator is an Expert System taking care of all tasks of analysis and programming. It does so in a completely automatic and transparent way.

The user states clearly ( in his own language ) the results he wants to obtain, and SOKRATES will help him through the Help function.

The Expert SOKRATES will detect and build automatically the relations, the structure of the files, the interaction in the treatment of informations, the tasks ( programmes ) and the windows for dialogues etc...

2.SOKRATES allows the completion of tasks in a competitive, recurring, multi-processing mode and in a network environment ( total multi-users ).

It manages the resources with logical flags and protects the integrity of the database through pre-imaging ( when in a transaction )

3. The application and the structure of the data can be modified, at any time, without any consequence on data already existing. SOKRATES takes care of its up-dating.

For SOKRATES the notions of size of record and of field do not exist. Therefore, fields can be of variable ( max.64 000 car.) or fix length.

# SOFTWARE TECHNOLOGY

Their presence in a file is optional or mandatory.

They can be of various type : data, sound, text, table (matrix).

4. It offers to developers a fully automatic choice of functions:

A. Construction of Windows and of print-out.

B. Restructuring of the database

- Fields of various length and non compulsory presence in the recording.

- Modification of the record's structure ( deletion, insert or modification of fields ) with no consequences on data already existing.

- Reorganisation of files when modifying access keys.

C. Execution of tasks in both multi-transactional and full-screen mode.

D. Function Keys : Exploration of files ( Scrolling, Zooming and Scanning ), and Help texts ( adaptable by the user ).

E. Execution of tasks : concurrent and recursive.

F. Management of resources in a network environment ( logical flags - full record-locking ).

G. Simultaneous dialogue of several users, on the same application, in their own language.

H. Protection and limitation of data access ( at all level : zone, recording, file, task, application... )

I. No limitation in the number of fields, files, tasks or data ( the only limitation is with the Hardware ).

# SOFTWARE TECHNOLOGY

- J. The optimisation of discs access has been treated with special care. It is very fast, for ressources of the RAM memory are used at their best ( hatching of libraries and cashing of files ).
5. With **SOKRATES** your applications can only be conceived through an approach "per object", with a treatment in structural mode ( Top-Down ).
6. SOKRATES produces a full documentation of applications and supplies totally automatically, a detailed analysis written and updated for each task.

*In all your applications, in all your computer works, SOKRATES will follow you to the limits of your imagination.*

# SOFTWARE TECHNOLOGY

## 2.INSTALLATION OF SOKRATES

In order to use SOKRATES on your own computer you need a PC system including:

- 1 MB RAM
- 1 Hard Disk with 2 MB available space
- 1 Floppy Disk Drive
- MS-DOS 2.xx or higher
- EGA or VGA Monitor with Adaptor and Monochrom
- Optionally, a Printer with Adaptor

the Config.SYS file in the root will include the following parameters:

Files=100 and Buffers=60

in order for SOKRATES to work properly.

You have received the system disk labelled:

SOKRATES  
Expert Applications Generator  
with  
Relational Information System

Release 1.xx

The system disk contains the programmes needed to install SOKRATES on your standalone computer.

Insert it in Drive A. Type : A:install , and press Enter key.

A few minutes later the installation will be over. You will notice that:

1.The following files have been created in the root directory:

BTRIEVE.EXE, BUTIL.EXE and GO.BAT.

# SOFTWARE TECHNOLOGY

2.A new subdirectory \SOK has been created containing the following files:

SOKRATES.EXE	Generator and Run-time
S.100.ST	Structure of Dictionary
S10001.ST	SOKRATES Help texts
BTRIEVE.EXE	Btree / ISAM . Index Management
BUTIL.EXE	Btree Utility / ISAM
GO.BAT	Batch file to start the program

Put the System Disk in a safe place. You are now ready to run SOKRATES by typing GO and Enter.

The next chapter is a quick tour of SOKRATES functions and performances. Reading the manual before starting using SOKRATES will help but if you like to explore,... go ahead!

At the first session you will have to indicate the initials of your installation : Three capital letters.

ex. Smith & Sons Ltd. : SMI = Identity code.

This will allow SOKRATES to personnalize your entire production with a suffix similar to the Id code of your installation.

You will notice that all the application dictionnaries and databases will have a suffix : SMI .

ex.: B100101.SMI or A100.SMI.

The first screen that will appear when the session starts concerns the choice or the implementation of a dictionary of applications.

You select the occurence : Installation ; Then you create the users, the companies ( databases ) and the applications.

NOTE : You should indicate a Mnemonic followed by the complete name of the users, companies or applications... as follows :

User : Alan DELAWARE = DEA: DELAWARE Alan  
Company : Combustible Corp. = COM: Combustible Corp;

# SOFTWARE TECHNOLOGY

As soon as you have created these elements, you quit the application : Installation with Ecs. and you select the application to process.

Do not forget to assign at least one company by application and at least one user in the company by application.

# SOFTWARE TECHNOLOGY

## QUICK-GUIDE FUNCTION KEYS

Abort, Exit, Stop or Ignore action in process	Esc
Application Help (Displays application help screen)	F1
Keyboard Help (With Ctrl, Alt and Shift)	F3
Sokrates Help	Alt F1
Adds one occurrence at end of list (append)	Alt F5
Cancels one occurrence in one list	F10
Cancels transaction in process	F10
Confirmation of transaction (accept)	Space
Choose a menu option or occurrence	Enter
End of transaction	Esc
Free Printing of a window (instant report)	Shift F2
Insert an occurrence in the list	F5
Possible answers ?	F2
Jump to previous field	Shift Tab
Jump to next field	Tab
Scan (access, elements, answers or tasks )	F2
Scroll Right	Ctrl F2
Scroll Left	Ctrl F1
Scroll to End	Ctrl F9
Scroll to bottom of page	Ctrl F7
Scroll to beginning	Ctrl F8
Scroll to top of page	Ctrl F5
Scroll down	Ctrl F6
Scroll up	Ctrl F4
Stack of tasks in process	Alt F2
Previous transaction	Alt F4
Next transaction	F4

All these function keys are self documented. Pressing F3 or Ctrl, Alt F3 and shift will prompt the list of function keys.  
By selecting with the cursor from the list of options and pressing Alt-F1 you will obtain the help text for that function.

# SOFTWARE TECHNOLOGY

### **3. THE EXPERT - SOKRATES**

The Expert-SOKrates builds and safeguards your applications within a Global Dictionary in two divisions : The Dico-Application and the Dico-Text (all texts and literal expressions). The Dico-Application is unique. The Dico-Text is common for all the different users languages.

These divisions are stored in the following files:

Annn.INS Dico-application with nnn = Dictionary number  
Annnll.INS Dico-texts ll = Languages number  
INS = Installation ID

The Expert-SOKrates (SOKrates.exe) uses an assistant (S.002) enabling it to interpret what it reads in the Dico-Application.

The Tasks execution ( exploitation ) generates and manages automatically the entities of the database. Each of these entities has a name or a format as follows:

Bdddccnn.ST, with cc equal to the company's number ( management entity, one database per company ), and nn equal to the number of the database's entity.

### 3.1. THE DICO-APPLICATION

The central dictionary of the application is composed by the following elements:

- |     |           |               |         |
|-----|-----------|---------------|---------|
| 1.  | Info      | (fields)      | Natural |
| 2.  | File      | (files)       | Real    |
| 3.  | Functions | (expressions) | Logic   |
| 4.  | Sheet     | (windows)     | Logic   |
| 5.  | Task      | (tasks)       | Logic   |
| 6.  | Print-out | (print-out)   | Logic   |
| 7.  | Paper     | (paper)       | Natural |
| 8.  | Font      | (font)        | Logic   |
| 9.  | Literal   | (constant)    | Logic   |
| 10. | Message   | (messages)    | Logic   |

# SOFTWARE TECHNOLOGY

Each element can be defined and updated independently. Management of the interaction with the other elements of the dictionary is fully transparent for the user, thus SOKRATES adapts itself to any approach and scenario of construction and implementation followed by the conceptor.

You identify an element by its name or by its Mnemonic. The mnemonic is always before the name (in full); It should not exceed 8 characters. The sign : is the separator:

Mnemo:Full name

Ex.: The Item ID. would be                   iteid:Item identificatio  
   Order date                           ordat:Order date

The mnemo is optional. It is recommended to use a mnemo for all infos names because of the convenience of use as operands or qualifiers in the functions, commands and evaluation expressions.

Never use the following characters in a full name or mnemonic:

# . & @ = ! / \* - \_ " (dont use space in mnemonics)

Each element of the dictionary is protected against unauthorised use.

There are 16 levels of authorisation to which each element is responsive.

Each user can receive several levels of authorization and during his sessions, the authorized objects for these levels will be accessible and operative.

SOKRATES allows only the conception of application in an approach " BY OBJECTS".

# SOFTWARE TECHNOLOGY

## 3.2 OBJECT APPROACH CONCEPT

With SOKRATES, both the Conception and Implementation of a Data system are done in one operation. The only prerequisite is to have a perfect knowledge of the problem to solve, i.e.:

- The results to establish ( their form and substance)
- The available data

Each Data, Fact, Algorythm, Rule or Formula et c... can qualify as element of the Dictionary. Each and all elements of a Data system has a private life and a social activity ( interactivity with other elements )

In the dictionary of applications, each element can be define by :

- Its identity
- Its features
- Its interface
- Its components

SOKRATES identifies three types of elements:

- Natural elements i.e. Data (value, sound, text, image,...)
- Real elements i.e. File (collection of group of Data)
- Logical elements i.e. All other objects.

The content of the real elements is in the relational Data system.

Contrarily to the natural element, the real or logical elements can be made up of other elements of all types.

When a real element is linked with other elements through generation, it shares all their resources through heritage.

An example will allow us to comprehend these notions more easily :

In a company's personnel Data files, we have for each employee the following natural informations :

Name, address, telephone, qualification, date of birth, photo, phonic curriculum vitae, et c...

# SOFTWARE TECHNOLOGY

The saving of the data concerning each employee initiates the creation of a new natural element : i.e. the Identification file of the employee ( Employees registration ).

The employee's activity in the company generates two other natural elements : The pay slip, the salary records of the employee.

The relation Employee - Salary enable each element to reach the Data of all the other elements.

## SOKRATES FONDAMENTAL LAWS

1. Each element has its free will, unalterable, within the Data system. No one can act on it from outside.
2. The Expert SOKRATES must supply to each element the ressources required for its socialization within the Data system.
3. When used, an element is bound to socialize and to follow the Expert's rules.

In short, an application developed with SOKRATES is formed of :

- A central Dictionary including the definitions of the elements of the Data system.
- A relational Data base where the real elements (files) of the Data system are stored.

The activity of the system consists in the execution of tasks where the recording of files are treated ( capture, evaluation, and cancellation ).

We must understand that at the beginning of a Data system there is the Data ( natural element ), that its capture and its evaluation are done within a task ( logical element ), and that its reality is in its presence within a file ( real element ).

# SOFTWARE TECHNOLOGY

The activity of a task requires the use of dialogues and print-out windows, of messages, of functions and of constants (logical elements).

### 3.3. INFOS

The info is the smallest element of information that can be identified in one application. From the developer's point of view it is the most important element of the dictionary and represents about 90 percent of the detailed analysis.

Similar to a living cell carrying the ADN code, the info carries a definition of its identity and an "evaluation code" (code ADN) that will be always executed each time the info is used (within a file, task, sheet, etc...).

The info also carries an expression of control of its content (test of plausibility) that is executed immediately after each input of the info.

One info can represent a part of another info (sub-chain). In this case it can be used like any other info. It can be inputed, evaluated, or even used as a key segment.

A info or sub- info can be included in the composition of the access-key to a file. In this case, any modification of a info's structure will always be detected by SOKRATES which will automatically start the reconstruction of the access concerned.

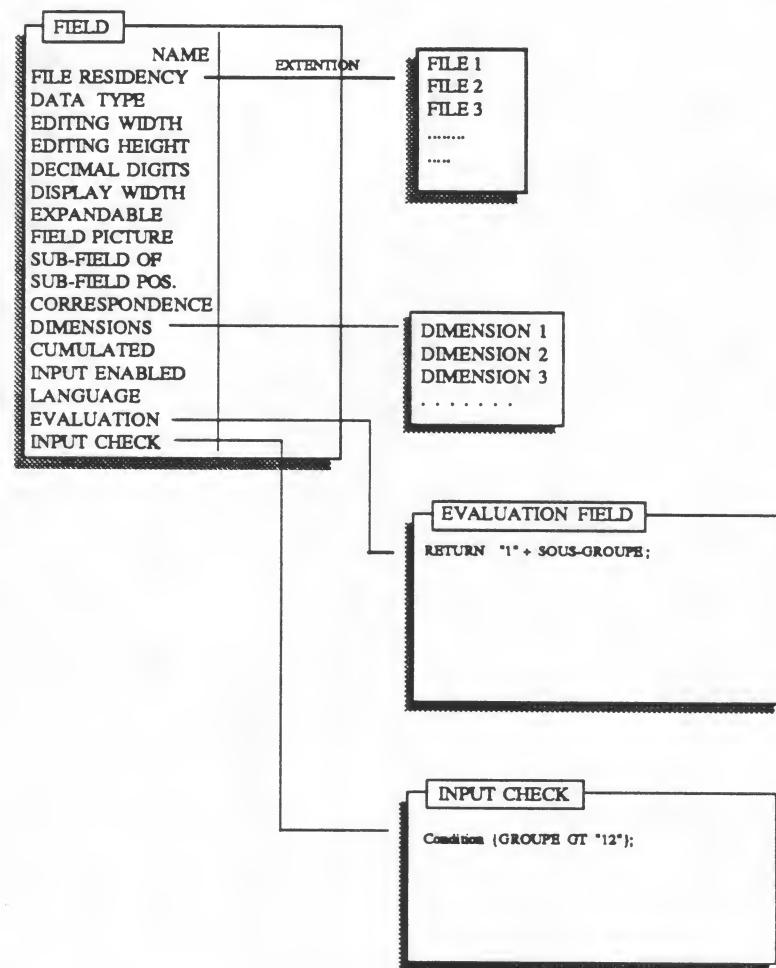
The info can be of fixed or variable length. Its format can be of the following type : date, data, text, hour, numeric, image, table (matrix). Its editing window could be different of the display window.

Every info receives the definition of its editing window (capture and process) and of its display window (out of edit mode). The dimensions of a window are the width (in characters) and the height (in lines).

Although the display indicated a height the info's edition, SOKRATES displays the info over one line only, unless the editing attributes of the info are stated.

# SOFTWARE TECHNOLOGY

## INFO STRUCTURE



USE HELP-KEY (Alt F1) FOR EACH ATTRIBUTE

# SOFTWARE TECHNOLOGY

One info could be authorised for input (within one or all the tasks) or not. It could be of the Instantaneous or Flux type (see Glossary) and could be edited in Latin, Arabic or Hebrew.

Remember that during the creation of the elements of the Dictionary you can get help by pressing: Alt F1.

## 3.3. FILE

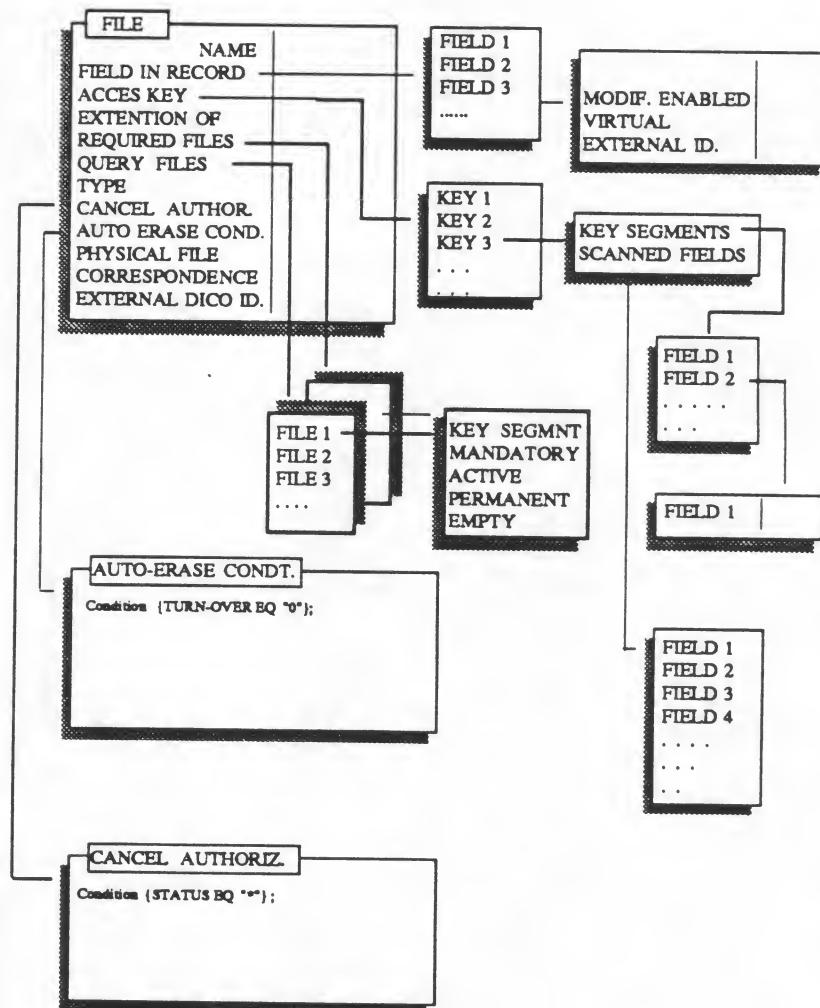
The file is a collection of infos associated by affinity of use and/or origin. In fact the file is a Macro-field that will be stored into a physical file. In order to find it, access keys must be defined.

The creation or modification of the contents of the infos of a file can only be done within the execution of a Task. The definition of the File includes:

- The list of infos (the sequence is not important).
- Access keys. The maximum number of permanent keys is 24. Each key can be composed of a maximum of 1024 non-contiguous segments. One segment can be a info or part of a field (sub-chain). One key can point to a record or to a file. The sequence of the keys is significant. The first key is always the primary key and it is necessary when using a SOKRATES file.
- One expression of evaluation of the cancellation condition of the records (keys) of the file and of their flux of Data.
- One expression of automatic cancellation of the records without erasing their flux of Data.
- The possibility for several files of different structure and access to reside (be stored) in the same physical file.
- The type of organization of the records and their access (FIFO, Sequential, Base, SOKRATES, etc..).
- The filiation of the file which can be the extention of another father-file.

# SOFTWARE TECHNOLOGY

## FILE STRUCTURE



USE HELP-KEY (Alt F1) FOR EACH ATTRIBUTE

# SOFTWARE TECHNOLOGY

- The type of residence of a file: permanent (in disk), in memory, temporary (existing only during task execution).

Each info of the file can be defined as a virtual info (not stored in the file). It will be calculated automatically by SOKRATES when first used in the task. An Info can also be declared as non modifiable when present in the file.

A info that is not a segment-key in a field can be a segment-pointer to a relation's file (boolean) (see Link element chapter).

During task execution, the Expert-SOKRATES can complete the list of infos of a file if it detects a relation that requires inexistant segment-pointers.

### 3.3.1. EXTENTION ("FATHER and SON")

A file can present in the list of infos an important repetition (sometimes indefinite) of similar infos. Example:

The order of a customer has several lines of details of the order. Each detail line includes the following infos:

Line Number  
Item code  
Item description  
Price per unit  
Discount  
Type of tax

It would be wrong to insert 50 times this infos in the calling list of the order file. What if the order had 60? or 2 lines?

It is a better method in this case to create a file "Heading of the order" (father) and a file "Detail of the order" (son). The second is an extention of the first. Consequently its name (primary key) will include the name of the father (primary key of the father=order #) and its "first name" (line #) that will differenciate him from his brothers. This is an extention father-son.

# SOFTWARE TECHNOLOGY

In case of wedlock where the father has a unique son, the case is solved by giving the son only the name ( and no first name). SOKRATES accepts polygamy. A father can contract several marriages and beget several sons.

Ties (links) of blood do exist in SOKRATES. The sons can be fathers and so on for many generations.

When executing the transactions of the sons, SOKRATES makes available the resources of the father or ancestors.

No one can go to the son without passing by the father! This means that the transactions of the son can not be processed without processing the father or the ancestors.

*Keep this in mind when designing your tasks!*

Do not confuse "the extention father-son" with the relation father-son (see Link element). The relation father-son is amore general relation where the keys of both files share a similar identity.

## 3.4 FUNCTION

When an expression of evaluation is used more than once it is recommended to list it and build it into a Function.

Its definition is composed of the details of the evaluation expression and its type:

- Internal : SOKRATES-Application
- External : External programmes in Pascal, Cobol,...

One function can contain operands that will qualify when called. Those are the formal (\$nnn) operands. In this case the function must have an input list of the variables of qualification:

Ex: Function for tax calculation  
Tax = (Qty \* \$1) \* \$ 2 / 100  
with \$1 = price and \$2 = tax rate

# SOFTWARE TECHNOLOGY

Calling the function will be:

Return @ Tax (unit price, VAT rate)

A function can be used as operand in an expression and can be called recursively.

The function is an expression of evaluation. Its execution always returns the result of the last condition operation and the value of the last Return or Init command ( see Return and Init commands) .

Logical operation : 1 = Right and 0 = "Wrong  
Algebraical operation : The result

## 3.5 . SHEET

The sheet is a window for dialogue and editing defined for the use of the operator or results. A sheet can be addressed to the screen, printer or any other terminal device. It can be automatic (built by SOKRATES) or manual (designed by the programmer).

Tree display modes for the info are available in the sheets:

### 1. Journal Mode (automatic)

The infos (title and contends) are displayed in columns. One info by line and by column. One parameter indicates the maximum number of lines per column. In this way the number of columns is automatically established according to the number of infos to edit.

The particularity of this mode is to display only one transaction (record) by window, the one currently in process.

Example: Maximum 5 lines / column

Code	: _____	Repres.	: _____
Name	: _____	Type	: _____
Address	: _____	Turnover	: _____
Phone	: _____	Margin	: _____
Telex	: _____	Last Sale	: _____

# SOFTWARE TECHNOLOGY

The default mode, when no sheet is specifiedm is journal mode with 1 column.

## 2. Transaction Mode (automatic)

The contents of the infos are displayed on the same line for each transaction. The titles are inserted in the frame built by SOKRATES. There is a transaction (record) per line and a Parameter will indicate the maximum number of lines (max. number of visible transactions).

This mode gives the user the ability to see several transactions of his file in the sheet and choose the one he wants to process, create or scroll the transactions of the file.

Example: Maximum 3 lines:

Code	Name	Address	Phone	Telex	Rep	T.O.

## 3. Manual Mode

When the sheet is designed by the concepor or end user, the display of info titles and their contents are up to him .

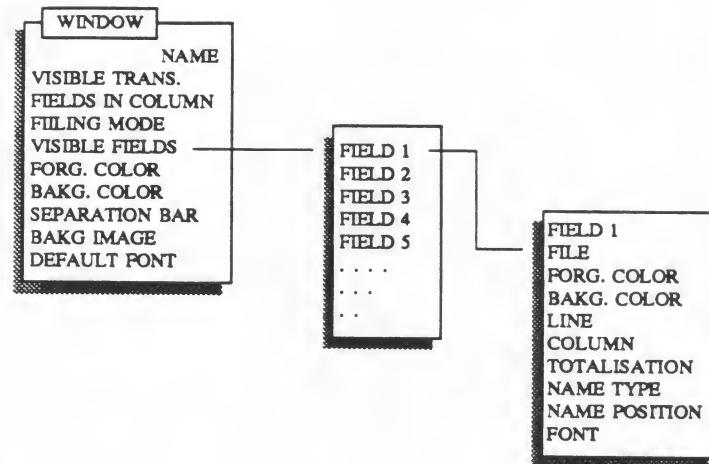
SOKRATES will make a space (height in lines and width in charaters) available and the operator will design the sheet details using the functions proposed by the system.

In the 3 modes, a window is always defined by the same parameters:

- List of infos used in the sheet with the editing attributes for each finfo
- Default editing attributes for the sheet.
- Topologic display coordinates.

# SOFTWARE TECHNOLOGY

## SHEET STRUCTURE



USE HELP-KEY (Alt F1) FOR EACH ATTRIBUTES

# SOFTWARE TECHNOLOGY

-Number of visible transactions or maximum number of lines per column (this parameter does not apply to Manual Mode).

-Height and width of the sheet ( max. 256 lines by 1000 columns).

-Visible sheet height and width.

## 3.6. TASK

The task element associates in its process of execution all the other elements of the Dictionary. It is the only concept known by the end user as all implementation of a Data Processing Application must lead to (result in) the execution of tasks specialised in the process of transactions.

A priory, its construction is nothing else than the statement of the elements it uses, their sequence and frequency of utilisation.

The parameters of a task definition are:

- The file processed in the task (one transaction = one record read and processed or to be processed).

- The list of infos processed in the task stated in the sequence of treatment ( Top-down). The sequence is significant.

- The sheet used (by default: an automatic sheet with one column journal mode )

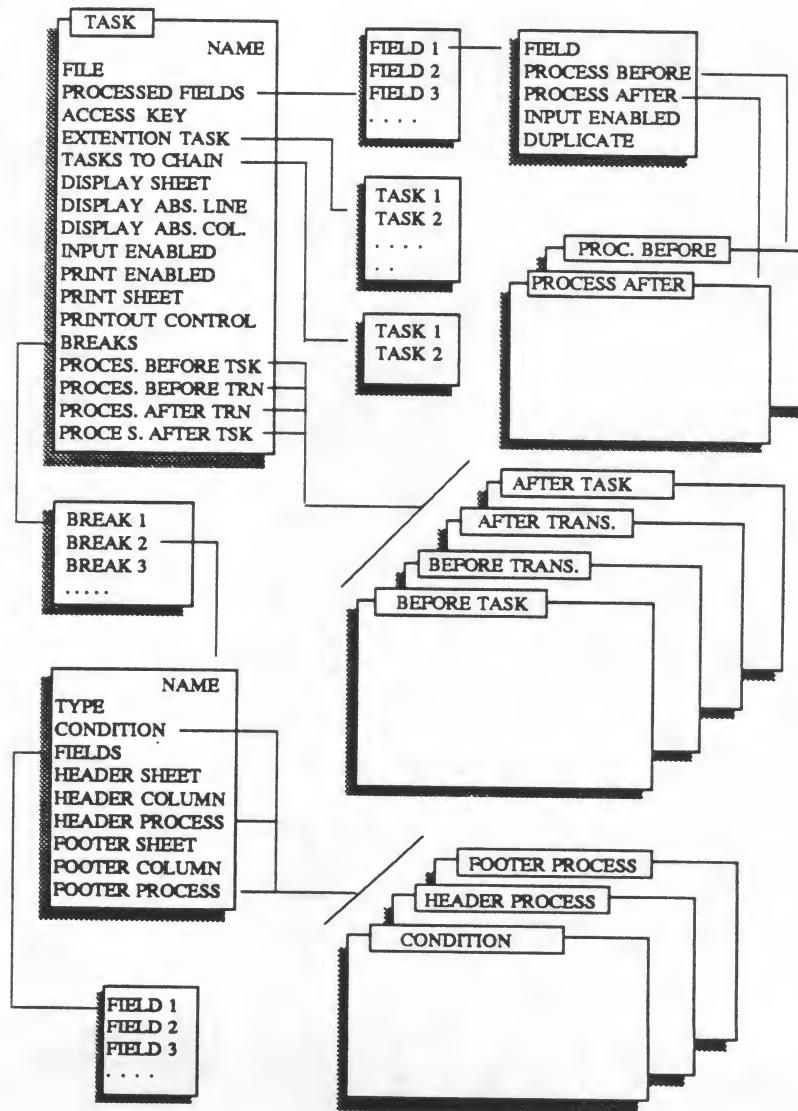
-The access key to the file's records. When in an interactive task SOKRATES always assumes the primary access.

- The list of father-son extention tasks.

- List of tasks to execute (chain) at the end of task (they will be proposed to the operator). If the list contains only one task it will be immediately executed.

# SOFTWARE TECHNOLOGY

## TASK STRUCTURE



# SOFTWARE TECHNOLOGY

- The printing configurator to use (for editing tasks only )
- Task of auto-start in case of "Pirate" or abnormal break of the process of treatment of the task.
- List of Control break in the priority sequence of the events releasing the Break. (see paragraph Control breaks).
- Other parameters are to be defined but before doing that we have to cover some basic theory:

## 3.6.1. WHAT IS A TASK ?

The fact of displaying a field, editing a field (input) displaying a window, inputting a transaction, printing a report, drinking a glass of water, driving a car, etc... is, indeed the expression of a task.

The process events of any task can be decomposed in several elementary micro-tasks that compose the macro-task and that are executed in a defined sequence.

In every task (micro or macro) one always finds the same steps, those of a slave system. They are:

- A. Before : Initialisation = Preparation
- B. During : Execution = Treatment
- C. After : Control = Retroaction

The phases of initialization and/or control may not exist in some cases. The phase of treatment must always exist, otherwise there is no task. But the execution could be "empty" (neutralised). Ex: The input of a field could be disabled.

In view of the preceding theory we can conclude that the inputting of fields (or treatment of fields) of a transaction within a task is nothing else than the nesting of micro-tasks, executed in the Top-Down sequence without field forward/backword jumps. We obtain in this way the General Schema of a Macro-Task explained in the coming page.

This schema is universal and represents the cornerstone of the whole

# SOFTWARE TECHNOLOGY

conception of computer based automatic information processing applications. It's indeed the case with SOKRATES!!.

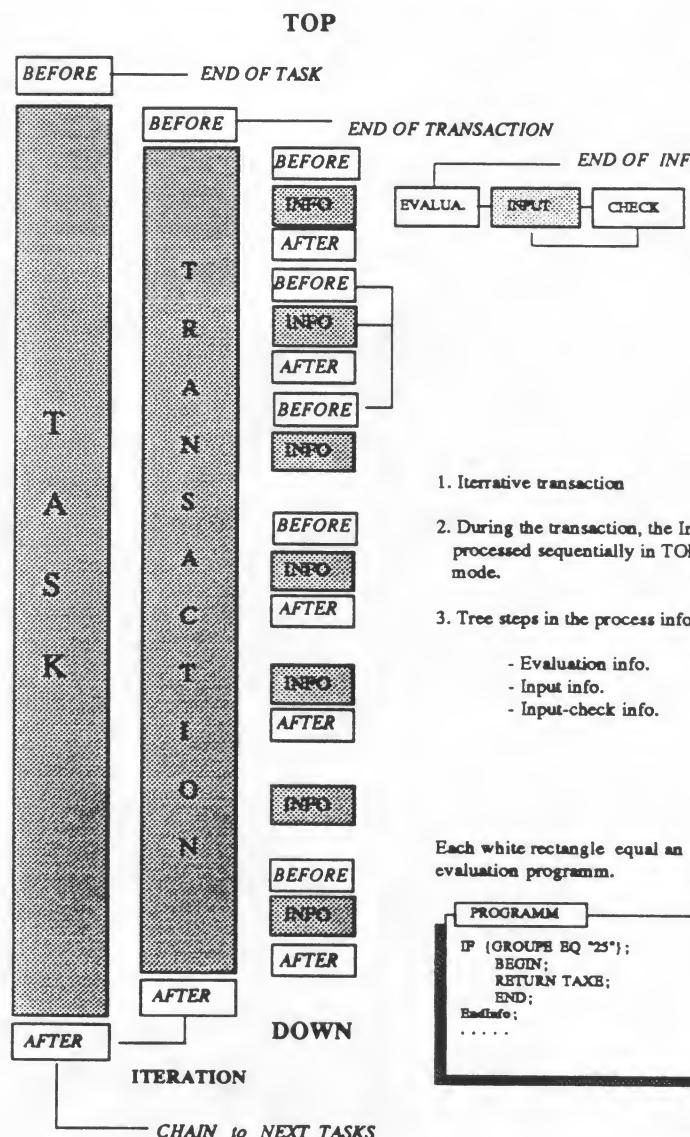
It is most important to understand well this schema and be able to apply it to the different real life scenarios.

From this comprehension will flow the state of the art architecture and the optimisation of tasks of your applications.

If the field is the most important element for the developer, the task and its windows are the most important from the operators point of view. The fields contain nearly the whole detailed analysis, while only the tasks are known and used by the end user.

# SOFTWARE TECHNOLOGY

## GENERAL SCHEMA OF A TASK



# SOFTWARE TECHNOLOGY

The expression programs-before/after means the execution of evaluation expressions containing logical or algebraic expressions, functions, and SOKRATES commands (see chapter : SYNTAX).

**IMPORTANT :** These programs cannot modify the contents of an Info such a modification can occur only as the result of the info's own evaluation or of its input.

As you can observe in the preceding schema:

- The program-before of the TASK will be executed only once at the beginning (entrance) of the task.
- The program-before of the TRANSACTION will be executed at the beginning (entrance) of every iteration of the transaction. ( after the transaction is red ).
- The program-before of the INFO will be executed immediately before entering into the field processing. It could cause the immediate processing of the following info.
- When processing the info, there is always an initialphase of valorisation of the info according to the expression of valorisation (in some cases the Expert-SOKRATES will not do it). Then, capture of the info will be proposed to the operator (if the input if allowed).
- Once the capture is over and SOKRATES has validated the operation, the program-after of the info is immediately executed. If capture is not validated, SOKRATES will position itself in the capture position again.
- At the end of transaction, and if it (the transaction) has been accepted by the operator and by SOKRATES, the program-after (the transaction) is immediately executed. It could request the end of transactions treatment, therefore the end of task. The operator can always control the end of the task if its type allows him to do so.
- The program-after of the task is executed only once at the end of the task. It could link the execution process of another task. Otherwise the task or tasks to chain will be activated.

Now we are able to complete the list of parameters of the task definition: We refer to the program-before/after DITO.

### 3.6.2. TYPE OF TASKS

In Data processing one often talks about batch processing, Editing programs, Menus, Interactive processing programs. In reality the Expert-SOKRATES only knows the concept of task which it defines in general as an INTERACTIVE task.

All other types of tasks are nothing else than particular cases of the interactive task.

#### DEFINITION OF THE INTERACTIVE TASK

This task has the capability to capture fields of a transaction requiring operator confirmation (acceptation). All modified transactions (records) will be automatically be written in the Database.

Now we can define the other type of tasks:

#### BATCH TASK

This task does not allow the capture (key-input) of fields and consequently no operator's confirmation is required. The transactions reading will be sequential ( following a choosen access-key) from the task's file and the updating of the modified records will be automatic.

#### PRINT OUT TASK

This task shares the same restrictions as the Batch task but the records updating is not allowed and the windows are addressed to the printer.

#### MENU TASK

This task shares the same restrictions as the Batch and Editing task but has no transaction file declared. The Expert-SOKRATES offers directly the list of tasks to chain, as it will detect immediately the end of file, hence the end of task.

# SOFTWARE TECHNOLOGY

### 3.6.3. CONTROLLED BREAKS

The process of task execution can be interrupted if certain events occur. The task will resume after the break.

The concept of Break is naturally introduced when defining the events responsible for the break of the process and the treatment to execute in connection with the break.

In SOKRATES, the definition of an event is expressed through an evaluation expression related to the break. If this expression is true, then the event happens, the process is interrupted, the treatment linked to this break is executed, and the process resumes.

Several events may take place and generate, for each of them, a specific treatment. In this case we would have as many breaks as events. Therefore, we must arrange these controlled breaks in a priority sequential order.

A break is considered as an extention process of the task process. Consequently it can have windows and "before and/or after" Break expressions.

In conclusion, a controlled break is defined by :

1. Program-event (condition)
2. Window (window)
3. Program-before (header)
4. Program-after (footer)

Within a task you can define as many breaks as you wish.

At the beginning of each new transaction SOKRATES detects automatically if at least one event takes place. In that case, all events of a lesser rank in the priority sequence of events will also be true and their treatment will be executed.

Let us take a sample case : The task's treatment process is to :

- Edit a customer sales statistics report,
- Establish a sales total by customer,

# SOFTWARE TECHNOLOGY

- Establish a sales total by commodity group,
- Establish a grand total.

The task "Sales Statistics" is created with the following definition:

File	:	Sales detail
Access	:	By customer and group
Input	:	Not authorised
Update	:	Not authorised
Field list	:	List of fields of the file
Window	:	Automatic or designed

Priority Breaks:

Final	(end of task)
Customer	(change of customer)
Group	(change of group)

It should be noticed that the final break (end of file or end of task) is always executed by SOKRATES without condition. Consequently, you do not have to specify any event (condition) for this level of break, but you may explicitly associate windows, headers and footers to it; and to the end of page break.

## 3.7. LINKS

The information available in the Database resides in several files. To access to the data and avoid redundancy of information, SOKRATES establishes links between the files automatically.

A file is a collection of related records. A record is identified by a key. One key is composed of segments included in the record.

A link is a relation established between 2 files. More precisely between one or several records of the calling file alone or several records of the called file.

In other words, the calling record contains the identification (the key) of the called record or records.

# SOFTWARE TECHNOLOGY

We shall see later on that the calling key can be composed of segments belonging to more than one record (case of a composed relation).

Within the following definitions, the word "key" always means the identification of a record, although it is often associated with an access key for reading the records of a file.

### 3.7.1. GENERAL RELATION

To a key of the calling file(start), can correspond none, one or many keys containing the calling key in the called file ( arrival ).

Ex.: Relation between the customer and order files.

File of start (calling) = Customers  
File of arrival (called) = Orders

For one customer, there can exist one, many or no orders at all. The calling key is the customer code and the called key must be the customer code (name of the father) AND the order number (first name of the son).

It is a relation that we shall call a father-son or extention relation. The father is the calling key and the sons are the called keys, as each contains the name of the father (calling key) and complementary segments (first name of son).

The reverse of the general relation that we shall call the mandatory relation is defined as follows:

To one calling key, must correspond one and only one called key. As the calling key represents the primary key of the called file, the order of creation of the keys is reversed: the called key must be created before the calling key.

#### Particular case:

If no field of the called record can be captured, then the mandatory relation implies the automatic creation of the called key if it does not exist ( automatic creation ).

A mandatory relation is always injective .

# SOFTWARE TECHNOLOGY

According to the above example one could say that for every order, one and only one customer should exist, and that the customer must exist priorly to the creation of the order.

The called key in a mandatory relation cannot be cancelled as long as at least one calling key exists.

### 3.7.2. BIJECTIVE RELATION

If both the calling and called keys are primary and identical keys, it is a bijective relation which will always be mandatory in both ways. Following that example, a new restriction could be introduced where one customer can not have more than one order in process. In this case the order file can only have one order per customer.

### 3.7.3. COMPOSED RELATION

If the calling key is composed of segments coming from more than one file, the relation is called composed. A composed relation is always mandatory

The Expert-SOKrates builds automatically all the relations of the Database. You can prescribe to him certain compulsory relations

### 3.7.4. RESSOURCE SHARING

In a Local network ( L.A.N. ) environment ( server and multi-user) the concept of record sharing is a must .

The sharing applies to a data called simultaneously by several users. Exclusive rights of recording will be granted to the first user as soon as he will have modified an existing data or introduced a new record.

The ressource is automatically made available for the next user as soon as it is freed. Access for reading ( consultation ) of the record is always available in all record sharing situations.

# SOFTWARE TECHNOLOGY

Ressource sharing is fully managed by the Expert - SOKrates with logical flags. The Expert-SOKrates fully manages the resource sharing by using logical flags.

## 3.8. PRINT-OUT

Each editing task may use a configurator to identify which paper to use, the margins ( top, bottom, left,...), automatic heading options, totals carried forward,...).

## 3.9. PAPER

The type of paper used by a configurator may be defined.

## 3.10. FONT

Fonts are selected in the editing window. By default SOKrates will use the default font of your printer.

## 3.11. LITERAL

Each literal or numerical constant used more than once within expressions should be introduced in that element of the dictionary. The sequences of control fonts defining the fonts of your printer will be defined as literals.

## 3.12. MESSAGES

That element includes several families of messages:

- 1 D.O.S. messages.
- 2 Expert - SOKrates messages.
- 3 Database Manager's messages.
- 4 Application messages ( created by the developer ).

Each message includes the following indication:

- A text (one line)
- An action code
- A beep command

Like any other element, a message can have its own help text.

# SOFTWARE TECHNOLOGY

## 4 . SYNTAXE

All evaluation expressions ( programme before, after, event,...) are written under the control of the Expert - SOKRATES

The programmes can include algebraic calculation, logical expressions, fonctions, literal expressions ( constants and commentaries ), SOKRATES and D.O.S. functions.

The syntax of SOKRATES has of a very reduced vocabulary of instruction sets classified in three categories:

- 1 Operators
- 2 Operands
- 3 Commands

The application developer writes in full or condensed form the expressions' commands (statements). Each command is ended by a ; that is a separating caracter. When the Esc.Key is pressed, the Expert SOKRATES controls and validates the syntax.

If the developer is not very familiar with the syntax commands of SOKRATES he can work in assisted mode. In this case he can select his answers in the syntax windows proposed by SOKRATES by pressing function key F2 and using the mouse or the keyboard keys. Presssing Esc. stops the assisted mode.

### 4.1. THE OPERATORS

SOKRATES proposes five types of operators:

- 1. Algebraic (Arithmetic)
- 2. Logical
- 3. Comparison
- 4. Database
- 5. Unary

#### 4.1.1. Algebraic Operators

+ Addition	/ Division
- Subtraction	e Exponentiation
* Multiplication	_ Remainder of division

# SOFTWARE TECHNOLOGY

## 4.1.2. Logical Operators

AND	And
OR	Or inclusive

## 4.1.3. Comparison Operators

EQ	Equal	GE	Greater than or equal
NE	Not equal	LT	Less than
GT	Greater than	LE	Less than or equal

## 4.1.4. Database Operators

&EOF	End of file
&EXK	Existance of one Key
&EXL	Relation empty
&TKC	Current task
&NEW	New transaction
&MFD	Modification of the value of a field
&MKY	Modification of the value of a key

## 4.1.5. Unary Operators

&ABS	Absolute value
&INT	Integer value
&RND	Round value
&NOT	Contrary value
&L	Left alignment
&R	Right alignment
&C	Center alignment
&UP	Convert to uppercase
&LW	Convert to lowercase
&MIN	The smallest
&MAX	The greatest

## 4.2. THE OPERANDS

The operands can be used in the algebraic (arithmetic) and logical expressions (as variables) or with the commands (as parameters).

# SOFTWARE TECHNOLOGY

SOKRATES reserves some words for himself. They can not be used as mnemonics or full name of an operand. They are the reserved constants and functions. ( See "Details of SYNTAX" ).

This is the list of operands proposed by SOCRATES :

- Fields
- Literals
- Reserved constants
- Functions
- Expressions
- %nnn

Only the field may be qualified by the file.

# SOFTWARE TECHNOLOGY

## 4.3. THE COMMANDS

Here is the list of commands proposed by SOKRATES. All the explanations on their use and writing conventions can be found in " Syntax details".

Code	Command description
DS	Display sheet
MS	Message display
ER	Erase file
CN	Cancel transaction
AS	Display attributes
AP	Printing attributes
CH	Chain to one task
*	Commentary
C	Condition
B	Begining of block
BW	Backwards reading direction
FW	Forward reading direction
CA	Call execution Task
E	End of block
;	End of command
EE	End of evaluation
EX	End of task
ET	End of transaction
EF	End of field
I	Initialize
R	Return value to caller
MF	Mandatory field input
NP	Next page (page break)
IF	If
W	While

# SOFTWARE TECHNOLOGY

## SYNTAX CONVENTIONS

The COMMANDS are defined in the coming pages according to the following writing conventions:

1. Group of syntax
2. Command name (title)
3. Action performed
4. Syntax format
5. One comment

Group of syntax	COMMAND TITLE
<u>Action</u> :	Definition of the action of the command.
<u>Format</u> :	Code parameter of the command;
<u>Comment</u> :	Comment and Sample of the use of the com

# SOFTWARE TECHNOLOGY

## QUICK REFERENCE GUIDE

### COMMANDS

Command Description	Codification
Display sheet	<b>DS</b> <i>Sheet, line, col, abs, i/t/f;</i>
Message display	<b>MS</b> <i>Message;</i>
File cancelation	<b>ER</b> <i>File;</i>
Transaction cancellation	<b>CN</b> <i>File, expr. select.;</i> (*)
Display attributes	<b>AS</b> <i>Field (window, frg, bkg, h, b);</i> (*)
Printing attributes	<b>AP</b> <i>Field (window, mode );</i> (*)
Chain to one task	<b>CH</b> <i>Task;</i>
Commentary	<b>*</b> <i>Commentary;</i>
Condition	<b>C</b> <i>(a NE b) AND (c EQ d);</i>
Beginning of block	<b>Begin;</b>
Backwards reading	<b>BW;</b> (*)
Forward reading	<b>FW;</b> (*)
Task execution	<b>CA</b> <i>Task;</i>
End of block	<b>E;</b>
End of command	<b>;</b>
End of evaluation	<b>EE;</b>
End of task	<b>EX;</b>
End of transaction	<b>ET;</b>
End of field	<b>EF;</b>
Initialize	<b>I</b> <i>(a - c) / y;</i>
Return value to caller	<b>R</b> <i>(b + c) * x ;</i>
Mandatory field input	<b>MI</b> <i>(# ON ou # OFF );</i>
Next page (page break)	<b>NP;</b>
If	<b>IF</b> <i>( a GE b ) or ( x + y ) LE z;</i>
While	<b>W</b> <i>( a GT b );</i>

(\*) Commands not enabled in 1.00 release.

# SOFTWARE TECHNOLOGY

## Algebraic Operators

+ - / \* e \_

## Database Operators

&EOF End of file  
&EXK Existance of one Key  
&EXL Relation empty  
&EXE Extension empty  
&TKC Current task  
&NEW New transaction  
&MFD Modified field  
&MKY Modified key

## Logical Operators

AND And  
OR or (inclusive)

## Comparison Operators

EQ Equal  
NE Not equal  
GT Greater than  
GE Greater than or equal  
LT Less than  
LE Less than or equal

## Unary Operators

&ABS Absolute value  
&INT Integer value  
&RND Round value  
&NOT Contrary value  
&L Left alignment  
&R Right alignment  
&C Center alignment  
&UP Convert to uppercase  
&LW Convert to lowercase

## Reserved Constants

#DAT System date  
#TIM system time  
#USR current user  
#APL current application  
#CPN current company  
#LNG current language  
#VER Run-time relaease  
#SER Serial number  
#BLK Black  
#BLU Blue  
#GRN Green  
#RED Red  
#MAG Magnenta  
#YEL Yellow  
#CYA Cyan  
#WHI white  
#EMP Double-strike  
#ENL Enlarged printing  
#CON Condensed printing  
#ESC Esc constant

## Operands

Field  
< File > Field  
Field [b, e, i] [...] ...  
“.....”  
#Reserved constant  
(expression)  
\$nnn  
&Fonction ( exp1, exp2,...)  
&Reserved fonction

# SOFTWARE TECHNOLOGY

Command

DISPLAY SHEET

Action : Allows to display a sheet in the task's main sheet .

Format : DS *Sheet (lin, col, AbsolutePosition, ET/EI/REM );*

with : lin = origin line #  
col = origin column #  
abs = screen relative origins, if no, relative to the  
window.

The window will be erased :

EI = at next info  
ET = at end of transaction  
REM = immediatly

Comment: The parameters: AbsolutePosition, ET, EI, REM are optional.

When lin and col are not indicated, the window display is justified at the top right of the screen and the abs parameter becomes useless.

If the abs is not present, the origin coordinates of the window are tight(relative) to the task window's origin.

If none of the parameters (ET, EI ) is present, the window will be remain on the screen until the end of the task.

If EI is specified, the window will be erased after capturing of the following field.

The origin coordanates of the window always refer to the top-left corner of the window.

# SOFTWARE TECHNOLOGY

Command	MESSAGE DISPLAY
---------	-----------------

Action : Allows the display of a message.

Format : MS    *name of message* ;

Comment:

The definition of the message in the Dictionary includes a severity code as follows:

- Warning
- Retry : Beep and retry the input of info
- End of transaction : Next transaction
- End of task
- Fatal : End of session

The message will always be displayed on the 25th line of the screen and calls for a help text by pressing the F1 key.

A message always includes a beep. Therefore a beep can be obtain by creating an empty message.

# SOFTWARE TECHNOLOGY

Command

**ERASE FILE**

Action: Allows the cancellation of a file (kill file), at the end of task.

Format: EF < company > File name;

Comment:

This command is executed only at the end of the current task.

If the task does not end normally, the file is not cancelled.

If the file's and company's parameters are not stated, they will take the current value.

# SOFTWARE TECHNOLOGY

Command

**TRANSACTION CANCELLATION**

Action : Causes immediate cancellation of one or several transactions and their data flux.

Format : CN    *file, expression of selection;*

Comment : All the parameter of this command are optional . If not stated they will take the current value.

The cancellation concerns only the selected transactions, those in which the expression of selection is true.

The cancellation of the selected transaction will take place only if the expression of cancellation attached to the file is true and if the transaction's relations are empty.

If the transaction has descendants (extentions father-son) it will be cancelled only if all its sons or descendant are cancelled.

If the parameter "file" is not specified, then it is question of the cancellation current transaction of the current task.

If the parameter "file" is specified, the cancellation will take place within a whole and SOKRATES will take in account the existing relation between the current task's file and the file of the transaction to cancel.SOKRATES will cancel all the transactions of the father-son relations.

(\*) This command is not active in the release 1.00.

# SOFTWARE TECHNOLOGY

Command

**DISPLAY ATTRIBUTES**

Action: Allows to modify the display attributes of a field within a window.

Format: AS *info ( sheet,frg,bkg,h,b )*;

with :  
  *info* = name of field  
  *sheet* = name of sheet  
  *frg* = fore-ground color  
  *bkg* = back-ground color  
  *h* = hight intensity  
  *b* = blinking

Comment: The *info* parameter is required, the others are optional.

If the *info*'s parameter is the only one indicated, then the *info* will be displayed with the original attribute of the task or sheet.

(\*) This command is not active in the release 1.00

# SOFTWARE TECHNOLOGY

Command

**PRINTING ATTRIBUTES**

Action : Allows the modification of the printing attributes of a info in a sheet.

Format : AP *Info ( wsheet mod );*

with : info = name of field  
sheet = name of sheet  
mod = character mode

Comment:

(\*) This command is not active in the release 1.00

# SOFTWARE TECHNOLOGY

Command

**CHAIN TO A TASK**

Action : Stops the execution of the current task and chains it to the execution of the indicated task.

Format : CH task;

with : task = name of task

Comment: If the parameter task is not specified and the current task :

- has a chained task (and only one) it will be immediately executed.
- has more than one chained tasks :
  1. the chained tasks will be proposed to the operator to choose if the task allows field capturing (Input Enable).
  2. The chained tasks are executed sequentially one after the other if the task does not authorises field capturing.
- has no chained task, then there is immediate return to the calling task.

# SOFTWARE TECHNOLOGY

Command

COMMENTAIRE

Action : Allows the documentation of the expressions details.

Format : \* *Comment*;

# SOFTWARE TECHNOLOGY

Operands

RESERVED CONSTANTS

This is the list of expert SOKRATES reserved constants. You can use them within your expressions.

#DAT	= System date
#TIM	= system time
#USR	= current user
#APL	= current application
#CPN	= current company
#LNG	= current language
#VER	= SOKRATES version
#SER	= Serial number

COLORS :

#BLK	= Black
#BLU	= Blue
#GRN	= Green
#RED	= Red
#MAG	= Magenta
#YEL	= Yellow
#CYA	= Cyan
#WHI	= white

# SOFTWARE TECHNOLOGY

Command	CONDITION
---------	-----------

Action : Return an expression's logical result ( True or False )

Format : C ( a GE b ) OR ( x NE b );

Comment: Several conditions can be stated in an expression. The expression's result will be True if at least one condition is True.

*N.B.: This command should only be used in the following types of expression :*

- Input check
- Autorisation of key cancellation
- Condition of key cancellation
- Condition of event ( Break )

With SOKRATES, conditions may be combined, but beware that the expression may lose its clarity and that complexity is mother of all the "Bugs"!

# SOFTWARE TECHNOLOGY

Command

**BEGINNING OF BLOCK**

Action : Delimits the beginning block's commands of a IF or WHILE.

Format : Begin;

EX. : Begin; → Beginning of block  
Command x;  
command y;  
.....  
End; → End of block

Comment:

# SOFTWARE TECHNOLOGY

Command

**BACKWARDS READING**

Action: Causes backwards sequential reading of the file (descending direction of keys), while the condition is True.

Format: BW *file, condition;*

Comment: The change of direction does not change the file's position in the current key.

# SOFTWARE TECHNOLOGY

**Command**

**FORWARD READING**

**Action:** Causes forward sequential reading of the file (ascending direction of keys), while the condition is True.

**Format:** FW *file, condition;*

**Comment:** The change of direction does not change the file's position in the current key.

# SOFTWARE TECHNOLOGY

Command

**CALL TASK**

Action : Interrupts the process in the current task, calls and executes the indicated task, and resumes treatment in the calling task.

Format : CA task, expression of selection;

with task = name of task to execute  
Expression = logical expression for selecting the transaction to process.

Comment : These command is equivalent to CALL EXEC TASK.

The execution of the called task is done as a whole because it is external to the current task.

If the called task is in relation father-son with the caller task, all the "son" transactions will compose the whole of transactions to process.

If a selection expression exist, the process will apply only to the transactions for which the expression is true.

In short, it is the execution of a task in a competitive mode.

(\*) The selection expression is not active in the release 1.00

# SOFTWARE TECHNOLOGY

Command

**END OF BLOCK**

Action : Delimits the end of a block of commands.

Format : End;

EX. : BEGIN;      —> Beginning of block  
      Command x;  
      command y;  
      .....  
      END;      —> End of block

Comment :

# SOFTWARE TECHNOLOGY

Syntax

**END OF COMMAND**

Action : Delimits the end of a command .

Format : ;

Comment : All command must end with a “ ; ”

# SOFTWARE TECHNOLOGY

Command

**END OF EVALUATION**

Action : Interrupts the current process of expression evaluation and continues processing the current info.

Format : EE;

Comment :

If the interruption takes place in the "program-before", the process is started when capturing the info.

If the interruption takes place in the "program-after", or in the info's evaluation expression, the next info is treated instead.

If the interruption takes place in the info's input-check expression, the control is done when capturing the field's content.

# SOFTWARE TECHNOLOGY

Command

**END OF TASK**

Action : Interrupts the processing of the current task.

Format : EX;

Comment :

This command interrupts the processing of the task and leaves it to chain it to the indicated task(s) in its task chain.

# SOFTWARE TECHNOLOGY

Command

**END OF TRANSACTION**

Action : Interrupts the process of the current transaction.

Format : ET;

Comment :

If the transaction has been modified and the task authorises to write it to the Database, then there is a request for transaction confirmation (in interactive) or an immediate writing (in batch).

This command when placed in the "program-before" of the transaction allows to select and/or to exclude transactions for treatment.

# SOFTWARE TECHNOLOGY

Command

**END OF INFO**

Action : Interrupts the treatment of the current info and starts treatment of the following one .

Format : EI;

Comment : It is equivalent to the interruption of the treatment of the current info:

- Interruption of the evaluation of the expression in process.
- Immediat jump to the next sequence-info: ie to the beginning of the "program-before" of the next info.

# SOFTWARE TECHNOLOGY

Command

---

INITIALIZE

---

Action : Proposes a value when creating an info.

Format : IN ( a + b ) \* y ;

Comment :

This command can only be used when expressing a info's evaluation .

The expression's result is assigned to the info only at its first capture or at his creation. If the info exists already the initialize command is not active.

# SOFTWARE TECHNOLOGY

## Operands

## OPERANDS

The following is the writing conventions for the operands used by SOKRATES:

Literal      " azap-125.36"

Constant    #DAT,#TIM,...(reserved constants)

Infos :      Direct      Name of the info  
                 Qualified    <File> Name of info  
                 Matrix     Name of info (d, a, i) :

with : [,,] = one by dimension

d = start index (default 1 )

a = stop index (default = Max.)

i = increment (default= 1)

Functions : Internal      &name of function (var1, ...., varn)  
                 External     &prog. name (info string)  
                 Unary        &up name of info

Expression: ( a+b-c )

Constant : name of constant.

Register : \$nnn

## Comment :

The name of an element of the Dictionary or any other parameter of a command ( when it is not significant) can be defined by an operand that contains it.

# SOFTWARE TECHNOLOGY

## Operators

## ALGEBRAIC OPERATORS

The list of algebraic operators proposed by SOKRATES is as follows:

- + = addition
- = subtraction
- \* = multiplication
- / = division
- e = exponentiation
- % = remainder of division (v =Alt 246)

# SOFTWARE TECHNOLOGY

## Operators

## DATABASE OPERATORS

The list of Database operators proposed by SOKRATES is as follows:

&EOF file	= End of file
&TKC task	= Current task
&EXK file	= Existance of one key
&EXL relation	= Relation empty
&EXE file	= Extension empty
&NEW file	= New transaction
&MFD field	= Modified field
&MFK key	= Modified key

### Comment :

These are in fact, unary operators on the Database.

# SOFTWARE TECHNOLOGY

Operators

**COMPARISON OPERATORS**

The list of comparison operators proposed by SOKRATES is as follows:

EQ	= Equal
NE	= Not equal
GT	= Greater than
GE	= Greater than or equal
LT	= Less than
LE	= Less than or equal

Comment :

# SOFTWARE TECHNOLOGY

---

Operators

---

**LOGICAL OPERATORS**

The list of logical operators proposed by SOKRATES is as follows:

AND = and  
OR = or inclusive

Comment :

# SOFTWARE TECHNOLOGY

---

Syntax

PARENTHESES ()

---

Action : It is a syntax character that indicates the beginning and the end of an expression.

Format : *(expression)*

with : expression = detail of expression

Comment :

# SOFTWARE TECHNOLOGY

Syntax

**QUALIFIER**

Action : It is a syntax character that qualifies the file name of the info .

Format : <file name>

Comment : The infos of the sheets can be qualified by the sheet attributes.

Ex. : <customer> address

# SOFTWARE TECHNOLOGY

Command

RETURN VALUE

Action : Returns the numeric value of the operation to the caller element.

Format : R ( a + b ) \* x ;

Comment:

For instance, If an expression of evaluation contains this command in a info, the result of the "R" operation will be assigned to the info when the evaluation expression is executed.

*N.B.: This command is active only in the evaluation expressions of infos.*

# SOFTWARE TECHNOLOGY

Command

**MANDATORY INFO INPUT**

Action: Forces the operator to key-input the info within the current transaction.

Format: MI (#ON or #OFF):

Comment:

The purpose is to force the operator to capture the content of a info.

*N.B.: This command is active only in the evaluation expressions of infos.*

The action of this command is only valid during the current transaction.

# SOFTWARE TECHNOLOGY

**Command**

**PAGE BREAK**

**Action :** Go to the next paper page. Execute a footer breaks and a header in the new page.

**Format : NP ;**

**Comment :**

# SOFTWARE TECHNOLOGY

## Command

## IF

Action: Give the executing condition of the following block of commands

Format: IF ( a GE b ) OR ( x NE b );

EX. : IF ( a GT b ) AND ( d EQ "12" );  
Begin;  
Return amount;  
display file's window  
End;  
Execute file correction task.

Comment: Using this command you will be able to build only simple conditions:

IF the logical expression is true:

THEN execution of the following command or block of commands.

The ELSE does not exist for SOKRATES. To get the equivalent of the ELSE you have to write another IF command with the opposite condition.

You can think that an IF - THEN is nothing else than a WHILE - THEN non iterative.

SOKRATES allows to nest several IF and/or WHILEs.

However, reading the expressions will loose clearness and it is well known that complexity is the mother of all "bugs"!

# SOFTWARE TECHNOLOGY

Command

**WHILE**

Action : Executes the instruction of the next expression as long as the condition is true.

Format : W ( a GE b ) AND ( z NE w );

EX. : WHILE ( a GT b ) AND ( d EQ "12" );  
Begin;  
Return amount;  
display account #;  
End;  
Execute task tt;

Comment :

The condition of the WHILE is always controlled at the beginning of the WHILE.

You can think that the WHILE is only like a IF - THEN iterative.

If the condition of the WHILE is true, the next block of commands is executed.

# SOFTWARE TECHNOLOGY

## 5. SCENARIO FOR AN IMPLEMENTATION

### 5.1 INTRODUCTION

There are many analysis and programming methods. Our aim is not to write a long study on analysis. It is to describe the best scenario to allow you to prepare the conception of the application to be directly apprehended by the Expert SOKRATES.

You must remember, that in no case, the Expert SOKRATES will replace your reasoning capabilities nor the knowledge you must have of the problem to solve. But in association with SOKRATES, it will help you to :

- Found the solid bases of your application
- Learn to master your software
- Ensure the permanent adaptation of the software to the needs of your management.

In day to day life, every one of us has to solve various and more or less complex problems. To do so we imagine and apply the solutions that appear to be the best adapted to our goals, desires, limitations and above all possibilities.

To solve a problem in good conditions, you must first expose it in good conditions. It might seem obvious, but how many good solutions have been given to false problems, and how many wrong solutions given to real problems.

The task of the analysis is to define the Result you want to obtain and the Ressources that you must dispose to get it.

Then come the phase of implementation which is the subject of this chapter.

With SOKRATES, the implementation of applications gives a free choice to the developper. There are three fundamental steps in the development of an application :

1. Knowledge of the problem to solve :  
Comprehension of the solution and

# SOFTWARE TECHNOLOGY

definition of the results to establish.

2. Implementation of the solution.

Construction of the tasks and of all the related elements and dictionary.

3. Run the operation.

Instruction of the operators and follow the operation.

The first step depends on the ability of the conceptor to understand the problem ( through interviews and situation analysis before automation ), and of his experience. SOKRATES can not help him there.

Similarly in the third phase, the only help that SOKRATES can give in the education of the operators is the help texts and the operator's manual : SOKRATES - GUIDE.

For the second phase, it is the subject of this chapter and covers all aspects of development, implementation and maintenance of the application.

In the following pages you will find exposed a general scheme for an application implementation with SOKRATES.

Always look for simple and clear formulas and expressions, remember that SOKRATES will write them in clear during the analysis automatic editing and that they will be read by neophytes.

N.B. : *SOKRATES gives five standard documents allowing the written preparation of the solution's detail in a form immediately available when implementing.*

**DON'T SKIP THE FIRST PHASE !!!**

It is the cause and origin of all the troubles met when starting the exploitation : wrong or incomplete results or false algorithms and formulas.

# SOFTWARE TECHNOLOGY

## STANDARD SCENARIO

The implementation of an application is done simultaneously during the phase of development.

We shall see the different steps of the recommended scenario to carry out the creation and the maintenance of an application.

You have started a session, chosen the application and you enter the option Generator.

1 Creation of the informations

2 Creation of the tasks

During this phase, you will have to create several objects : the files, the sheets, the breaks, the print-out, the papers, the messages and the literal constants.

3 Creation of the possible mandatory links between certain files of the information system.

4 From the Root-task, compose the entrance main menu of your application.

5 Choose the occurrence Compilation of the links. SOKRATES will create automatically all the relations between files of the information system.

6 Choose the occurrence Information System to enter in the phase of tasks execution.

7 While executing the tasks, you can update all the help texts of the Infos, tasks and messages.

# SOFTWARE TECHNOLOGY

## 6. GLOSSARY

### Blok of commands :

It is group of commands defined the commands of beginning and end of bloc. The commands of a bloc are executed in a sequence from top to bottom.

### Calling key :

The calling key is composed of segments from one or several records. The segments of a calling key are defined as pointer-segments.

### Main key :

It is a group of segments ( one or several fields ) of the record constituting its unique identification ( not duplicable ). It is always the first key to be defined in the records.

### Secondary key :

It is a group of segments ( one or several fields ) of a record constituting its other access keys through different ways than the main key. The secondary key is duplicable, hence redundant.

### Empty command :

A commentary is always attached to a command ( after the ; ). To insert a commentary without an active command, you will use an empty command to which a commentary you attach a commentary.

### Dimension :

A field can be define as a board in several dimensions : Page, line, col., ...etc... Each dimension is limited by the maximum number of occurrences it can admit.

### Extention father-son :

If a transaction ( father ) has a group of repeated similar fields, these are assembled in another transaction ( son ). The son-transaction's main key will be formed of the father's main key and of one or several segments allowing the unequivocal identification of the son.

# SOFTWARE TECHNOLOGY

## Empty extention :

An extention is said empty when there is no son in a father-son extention.

## Only son :

An only son in an father-son is a son that has no first name. In other words, his main key is identical to his father's main key.

## Fonction :

A fonction is an expression of evaluation and of calculation. It is formed of algebraic and logic expressions and of commands.

It is said intern when its content is built with SOKRATES'syntax elements.

It is said extern, when its execution is done outside of SOKRATES and where he receives only a final string.

## Bijective Relation :

If the calling key is unique and is the file's main key, it is a bijective relation always mandatory in the direction of its surjection.

In the example of the general relation, a restriction must be introduce : a client can not have more than one current order. In this case, the file can not contain more than one order per client.

## Composed relation :

If a calling key is composed of segments from more than one file, the relation is said Composed. A composed relation is always surjective.

## Mandatory relation :

If the calling key is the main key of the called key, the relation is mandatory. In this case there is precedence in the creation of the keys : the called key must be created before the calling key.

# SOFTWARE TECHNOLOGY

N.B.: If the called record has no available field the obligation of precedence implies the automatic creation of the called key if it does not already exist ( automatic creation ).

A mandatory relation is always surjective.

In the example of the general relation, one should say that for each order must exist one and only one client that must exist when the order is created.

You can not cancel the called key of a mandatory relation for as long as at least one calling key exists. A mandatory relation is always surjective and implies creating the called key before the calling key.

## General relation :

To a key of the calling file ( origin ) can correspond in the called file, none, one or several keys containing the calling key.

Ex.: Relation between files Clients and Order.

Origin file ( calling ) = The clients  
Destination file ( called ) = The orders

To one client can correspond one or several orders, or none. The calling key is the client's code and the called key must be the client's code ( father's name ) and the order number ( son's first name ).

It is an injective relation that we shall call : father-son relation. The father is the calling key and the sons are the called keys, as each of them contains the father's name ( calling key ) and additional segments ( son's first name ).

We call the reverse of the general relation Surjective relation and it is defined as follows :

To one or several calling keys, can correspond none or one and only one called key.

# SOFTWARE TECHNOLOGY

Ex.: From the earlier example, to each order can correspond only one and only one client.

#### Injective relation :

An injective relation is defined as follow : to a calling key, can correspond none, one or several called keys.

The reverse of an injective relation is a surjective relation.

#### Father-son relation :

A father-son relation is defined as a general relation ( see general relation).

#### Surjective relation :

A surjective relation is defined as follow : to a calling key, can correspond none or one and only one called key.

The reverse of a surjective relation is an injective relation.

#### Empty relation :

A relation is said to be empty when no called key correspond to the calling key.

#### Controlled break :

The execution process of a task can be interrupted, in certain circumstances, and resume.

The concept of break is naturally introduced with the definition of the event responsible for the process' break and the treatment to execute following the break.

SOKRATES includes the definition of an event within an evaluation expression related to the break. If that expression is true, then the event has occurred, the process is interrupted, the treatment attached to the break is executed, and the process resumes.

# SOFTWARE TECHNOLOGY

Several events can occur and generate for each of them a specific treatment. In this case we shall have as many breaks as we have of events. Therefore, these breaks will have to be classified according to priorities ( following a sequential order of importance. )

A break is considered as an extention task of the process in which it occurs. It can have a window, a programme before and/or a programme after.

#### Key-segment :

A key-segment is one of the segments constituting a key. Its position in the key is significant for the selection order of the keys.

A key-segment can be formed of a field or a sub-field, both real.

#### Pointing-segment :

Similar to the key-segment, the pointing-segment is part of the constitution of the calling keys.

#### Sub-field :

A sub-field defines part of a field. It can be part of the keys and receive a value attributed by a calculation command.

A sub-field can not enter in the composition of a list of fields in a file nor in a task. However, it can be part of the list of fields of a windows.

**Task :** A task is a process of treatment that can be preceded by an initialization phase and followed by a control phase, or retroaction.

#### Batch task :

A batch task is an interactive task unable to capture fields, and consequently, exempt of confirming the treatment of the transaction. The file's transactions of the task are read in a sequence following the chosen code.

#### Editing task :

The editing task is a batch task not allowed to write the modified transactions. The tasks windows can be directed to the printer.

# SOFTWARE TECHNOLOGY

## Interactive task :

An interactive task is defined by the following features :

- 1 Autorized capture of seizable fields.
- 2 Confirmation of the transaction's treatment before writing the modified transactions.
- 3 Writing of the modified transactions, when the transaction's treatment is accepted.

## Task menu :

A task menu has no signed file and consequently is not concerned with any the rules of the other types of tasks. Its particularity is to have a list of chaining task. If the capture is autorized, then the operator is given the choice of the chaining tasks.

If the capture is not autorized, all the tasks of the chaining list will be executed in sequence. The chaining of each of these tasks is invalidated, and at the end of the list, return to the main menu.

## Flux-field :

The content ( its numerical value ) of a flux field is the result of the operation of addition of input and output flux of the database's transactions.

A flux-field is always numerical and can not receive a value by algebraic allocation.

## Instantaneous field :

On the contrary of the flux-field, the instantaneous field receives its value only by algebraic allocation.